resource identifiers such as a current date using resource functions into a localized format.

Turning next to Fig. 4, that figure shows a flow chart of a process for providing resources adapted to a user environment consistent with methods, systems, and articles of manufacture consistent with the present invention, and highlights the steps performed by the resource program. The steps described below may be executed using the resource program on the data processing system 100, however, the process shown in Fig. 4 is not limited thereto.

The process depicted in Fig. 4 provides internationalization support to applications that are written to be independent of a language or format desired by a particular user. Functions are provided that support provision of localized text resources, that convert locally different user inputs, such as a current date, into internal formats, and that convert internal formats into localized formats.

In step 21, the resource program sets a user parameter for selecting a user environment. The user parameter may be received from a user and may be stored, for example, in memory 130 as an object, such as a session object or application object, by the user parameter component 181, as described above with reference to Fig. 1. The user parameter may correspond to a user environment including a user location or a user language preference.

In step 22, an application that is independent of the selected user parameter is executed, for example, by the application component 182, as described above with reference to Fig. 1. The application may be controlled by the user, for example, for preparing frames for display on the video display 150 at the user's location. The frames can include, for example, graphic elements and text elements.

In step 23, the application component 182 reads a resource identifier from the application and that is independent of the user parameter. For example, the application executed by the application component 182 may be a program in the form of Microsoft® Active Server® Pages (ASP) for presenting information to a user.

In step 24, the lookup component 183 loads a lookup object from the lookup table 188 for linking the resource identifier, which is independent of the user

parameter, such as an identifier in an internal format, and resource data, which is dependent on the user parameter. The lookup object can be loaded from a session object or application object. The resource program may execute step 24 once upon entering a session with a user or may execute step 24 when a request is received from a user or when a user parameter is set by the user. The resource data may comprise a string of at least one character, such as an expression in a particular language. Further, the resource data may correspond to a resource function that includes rules for character representation, such as for date representation, time representation, currency representation, or floating point representation, as described above with respect to Fig. 1.

In step 25, the lookup component 183 calls a lookup function for obtaining resource data from the lookup object based on the read resource identifier. The obtained resource data may be presented to the user. For example, the resource data could represent an expression that is presented to the user in a language that is understood by the user, based on the selected user parameter. Alternatively, the resource data could represent a date or floating point number that is presented to the user in a format that is based on the user parameter. A plurality of lookup operations could be executed, when an application includes a plurality of resource identifiers.

The functions for initializing and loading the lookup object, as well as the functions for performing each of the steps of the process depicted in Fig. 4, can be written in a scripting language, such as the Visual Basic® scripting language.

Further, one of skill in the art will appreciate that the steps outlined in Fig. 4 do not necessarily have to be performed in the depicted order. Instead the order of the steps may be varied as appropriate. For example, step 24 may be executed at any appropriate point in time during execution or may be executed once at the beginning of a session.

Therefore, after the resource program sets the user parameter, the lookup component 183 can load the lookup object, including text resources and resource functions required for the application. The lookup object that is loaded for the application can include specific expressions or signs that are presented in connection with a particular application in different languages or formats, based on

the user parameter. It is also possible that a single dictionary, such as a lookup object, is used for all applications containing all resource data which need to be presented to a user.

After the resource program stores the dictionary, for example in a session object, e.g. identified by an object identifier or language identifier, the application, e.g. written in the form of the Microsoft® Active Server® Pages format can use a function which takes a resource identifier as a parameter and returns the resource text, i.e. the lookup function.

The lookup component 183 may further execute a lookup function to determine whether a dictionary exists for the selected user parameter and may retrieve a resource data or resource function from the lookup object. In case a resource identifier does not exist, the lookup component 183 can present an error message to the user that indicates that the resource identifier was not found.

Further, in a case where a resource function is used, user specific input, such as date, time, or floating point numbers, may be converted from a locally dependent format into a locally independent object. For example, the user specific input may be converted into a Visual Basic® Script object. This object that is independent of the selected user parameter may then be used for processing by the application. A result of the processing can then be presented to a user as resource data that is in a format or form corresponding to the user, that is, dependent on the user parameter.

When a number of N users is accessing the resource program, a lookup object, such as resource data and resource functions, may be stored for each user, for example, in a session object or application object. In this case, the resource data are stored N times, one time for each user in a particular language or format. Also, in this case, after a user exits the system, the lookup object may be erased. Further, the resource program can load a set of lookup objects for each user.

Further, for N users that are associated with M languages, the resource program can store one lookup object for all N users. In this case, the resource program stores resource data M times, i.e., in the languages/formats chosen by the users. Further, the resource program can load a set of lookup objects for the